

## Amendments to Claim:

This following listing of claims will replace all prior versions, and listings, of claims in the application.

## Listings of the Claims:

1. (currently amended) A computer implemented method for tracing an execution path through application components distributed over at least one computer in a data processing system, comprising ~~the steps of:~~

providing a plurality of interceptors installed in a plurality of application components, wherein each application component has at least one interceptor;

receiving a plurality of requests to said data processing system, wherein processing each request by said data processing system produces an execution path traversing at least one application component;

~~generating a tracing token, activating data collection, and relaying said tracing token in one of said application components distributed over at least one computer, and~~

for a first interceptor, intercepting a first operation at a first application component;

for a first interceptor, determining if tracing an execution path associates with said first operation is desired;

for a first interceptor, if tracing said execution path associates with said first operation is desired, performing a first trace function, wherein said first trace function comprises:

generating a tracing token;

activating data collection at said first interceptor; and

relaying said tracing token;

~~detecting said tracing token, activating data collection, and relaying said tracing token in at least one of said application components.~~

for a second interceptor, intercepting a second operation at a second application component;

for a second interceptor, detecting said tracing token; and  
for a second interceptor, if said tracing token exists, performing a second  
trace function, wherein said second trace function comprises:  
activating data collection at said second interceptor; and  
relaying said tracking token.

2. (original) The method of claim 1, wherein said application components include at least one of application programs, plug-in programs, program extensions, static library programs, dynamic library programs, program scripts, and individual modules, or subroutines, or functions, or procedures, or methods thereof.

3. (original) The method of claim 1, wherein said application components reside in the same computer, or are distributed over multiple computers interconnected by at least one of a local area network, a wide-area network, a wireless network, or the Internet.

4. (currently amended) The method of claim 1, wherein ~~said generating a tracing token is controlled by~~ said determining if tracing an execution path associates with said first operation is desired utilizes a start rule and a stop rule.

5. (currently amended) The method of claim 1, wherein said tracing token ~~[[is]]~~ comprises a unique value in said data processing system, said unique value uniquely identifies a particular request.

6. (currently amended) The method of claim ~~[[1]]~~ 5, wherein said unique value ~~includes~~ is a universally unique identifier (UUID).

7. (original) The method of claim 1, wherein said relaying of said tracing token is accomplished through at least one of reading and writing of a shared memory, attachment to an inter-process message, and insertion into a data stream.

8. (currently amended) The method of claim ~~[[7]]~~ 1, wherein ~~said attachment to an inter-process message includes at least one of a non-intrusive modification of the message header, a non-intrusive modification of the message body, or~~

appending to the message said relaying said tracing token step at said first interceptor comprises inserting said tracing token in an inter-process message, said detecting said tracing token step at said second interceptor comprises reading said inter-process message to retrieve said tracing token, and said tracing token is detected when said tracing token is found in said inter-process message.

9. (currently amended) The method of claim [[8]] 1, wherein ~~said non-intrusive modification of the message header includes at least one of adding a new field in a SOAP transport header over HTTP, or adding a new property in a SOAP transport header over JMS~~ said relaying said tracing token step at said first interceptor comprises writing said tracing token to a data stream, said detecting said tracing token step at said second interceptor comprises reading said data stream to retrieve said tracing token, and said tracing token is detected when said tracing token is found in said data stream.

10. (currently amended) The method of claim [[8]] 1, wherein ~~said non-intrusive modification of the message body includes the insertion of a comment in an SQL statement~~ said relaying said tracing token step at said first interceptor comprises storing said tracing token in a shared memory location accessible by said second interceptor, said detecting said tracing token step at said second interceptor comprises reading said share memory location to locate said tracing token, and said tracing token is detected when said tracing token is found in said shared memory location.

11. (original) The method of claim 1, wherein said tracing an execution path through application components takes place in at least one of system production time, system development time, system integration time, or system testing time of said data processing system.

12. (original) The method of claim 1, wherein said generating a tracing token is automatically triggered by detection of a system event.

13. (original) The method of claim 12, wherein said detection of a system event is dynamically activated and deactivated.

14. (original) The method of claim 13, wherein said dynamic activation and deactivation of said detection of a system event are controlled internally in said one of said application components, or externally to said one of said application components.

15. (original) The method of claim 12, wherein said system event includes at least one of an event relating to system and application performance, system and application availability, system and application troubleshooting, or system and application security.

16. (currently amended) A computer implemented method for tracing an execution path through application components distributed over at least one computer in a data processing system, comprising ~~the steps of:~~

installing first monitoring code in a first plurality of said application components for detecting system events;

installing second monitoring code in a second plurality of said application components for detecting tracing tokens, wherein said tracing tokens are generated by said first monitoring code, and said second monitoring code is different from said first monitoring code;

detecting one of said system events, activating data collection, and generating and relaying one of said tracing tokens in one of said first plurality of said application components, wherein said one of said tracing tokens comprises a unique identifier uniquely identifies said execution path;

detecting and relaying said generated one of said tracing tokens to activate data collection in at least one of said second plurality of said application components along said execution path through said application components in said data processing system; and

recording said collected data ~~[[in]]~~ from all said application components along said execution path through said application components in said data processing system.

17. (original) The method of claim 16, wherein said first plurality of said application components includes entry points for client enquiries to an n-tier client/server distributed computing system.

18. (original) The method of claim 16, further comprising the step of providing storage means for said collected data for all said application components along said execution path through said application components in said data processing system.

19. (currently amended) The method of claim 18, wherein said storage means includes at least one of files, registries, relational databases, [[and]] or object databases.

20. (currently amended) The method of claim 16, wherein a record of said collected data includes at least one of:

- the value of one of said tracing tokens;
- the value of a computer's timestamp;
- the identifier of a computer;
- the identifier of one of said application components;
- the description of one of said system events; or
- application data from operation.

21. (currently amended) The method of claim 16, wherein said monitoring code is installed in at least one of [[the]] source code, [[the]] object code, [[the]] executable code, [[and]] or application extension mechanism of said application components.

22. (currently amended) A computer program product comprising~~[[ :]]~~ a computer usable medium having computer readable program code means embodied therein for tracing an execution path through application components distributed over at least one computer in a data processing system, the computer readable program code means in said computer program product comprising:

computer readable program code means for installing first monitoring code in a first plurality of said application components for detecting system events;

computer readable program code means for installing second monitoring code in a second plurality of said application components for detecting tracing tokens, wherein said tracing tokens are generated by said first monitoring code, wherein said second monitoring code is different from said first monitoring code;

computer readable program code means for detecting one of said system events, activating data collection, and generating and relaying one of said tracing tokens in one of said first plurality of said application components, wherein said one of said tracing tokens comprises a unique identifier uniquely identifies said execution path;

computer readable program code means for detecting and relaying said generated one of said tracing tokens to activate data collection in at least one of said second plurality of said application components along said execution path through said application components in said data processing system; and

computer readable program code means for recording said collected data [[in]] from all said application components along said execution path through said application components in said data processing system.

23. (new) The method of claim 1, wherein said first application component and said second application component are the same.

24. (new) The method of claim 1, wherein said first application component and said second application component are different.

25. (new) The method of claim 24, wherein said first application component and said second application component run in the same process on a computer.

26. (new) The method of claim 24, wherein said first application component runs in a first process and said second application component runs in a second process, and said first process and said second process are different.

27. (new) The method of claim 26, wherein said first process runs on a first computer and said second process runs on a second computer, and said first computer and said second computer are connected to a network.

28. (new) The method of claim 1, wherein said first interceptor intercepts a function call of said first application program.

29. (new) The method of claim 1, wherein said first interceptor intercepts a function call of an operating system.

30. (new) The method of claim 1, wherein said second interceptor intercepts a function call of said second application program.

31. (new) The method of claim 1, wherein said second interceptor intercepts a function call of an operating system.

32. (new) The method of claim 1, wherein a decision of whether to trace an execution path is made by said first interceptor.

33. (new) The method of claim 1, wherein said first interceptor initiates tracing of an execution path.

34. (new) The method of claim 1, wherein data collected at said first interceptor and data collected at said second interceptor are stored in a database.

35. (new) The method of claim 1 further comprising:  
for a third interceptor, intercepting a third operation at a third application component;  
for a third interceptor, detecting said tracing token; and  
for a third interceptor, if said tracing token exists, performing a third trace function, wherein said third trace function comprises:  
activating data collection at said third interceptor; and  
relaying said tracking token.

36. (new) A method of selectively tracing application program execution paths comprising:  
providing a plurality of interceptors installed in at least one application program;

at a first interceptor, controlling start of tracing an execution path;  
at a first interceptor, when said start of tracing said execution path is granted, performing a first trace function at said first interceptor, wherein said first trace function comprises:  
generating a tracing token, wherein said tracing token comprises a unique identifier;  
activating data collection at said first interceptor; and  
transmitting said tracing token to at least one interceptor on said execution path;  
at a second interceptor, receiving said tracing token; and  
at a second interceptor, when said tracing token is received, performing a second trace function at said second interceptor, wherein said second trace function comprises:  
activating data collection at said second interceptor; and  
transmitting said tracing token to at least one interceptor on said execution path.

37. (new) The method of claim 36, wherein said controlling start of tracing said execution path at said first interceptor further comprising:

providing said first interceptor having a trace variable;  
examining said trace variable; and  
starting a trace when said trace variable is set.

38. (new) The method of claim 37 further comprising:

monitoring at least one system event at said first interceptor; and  
for each monitored system event, performing the following:

when said trace variable is not set, evaluating at least one start rule on said monitored system event and setting said trace variable according to the outcome of said evaluation of said start rule; and

when said trace variable is set, evaluating at least one stop rule on said monitored system event and setting said trace variable according to the outcome of said evaluation of said stop rule.



39. (new) The method of claim 36, wherein said first interceptor and said second interceptor run in a process on a computer, and said tracing token is transmitted from said first interceptor to said second interceptor via shared memory.

40. (new) The method of claim 36, wherein said first interceptor runs in a first process and said second interceptor runs in a second process, and said first tracing token is transmitted from said first interceptor to said second interceptor via inter-process communication.

41. (new) The method of claim 36, wherein said first interceptor is installed in a first application program and said second interceptor is installed in a second application program.

42. (new) The method of claim 41, wherein said first application program runs on a first computer and said second application program runs on a second computer.

43. (new) A method of selectively tracing application program execution paths comprising:

- monitoring response time on servicing a request at a first interceptor;
- controlling tracing an execution path on subsequent invocations of said first interceptor using at least one response time measurement at said first interceptor, wherein setting a trace variable on said first interceptor when tracing said execution path commences on subsequent invocations of said first interceptor and resetting said trace variable on said first interceptor when tracing said execution path does not commence on subsequent invocations of said first interceptor; and
- tracing a particular execution path of a particular request when said trace variable on said first interceptor is set, wherein said tracing step comprises:
  - for said first interceptor, generating a tracing token, collecting data into a database, and relaying said tracing token, wherein said tracing token comprises a unique identifier uniquely identifies said particular request; and
  - for a second interceptor, receiving and relaying said tracing token, and collecting data into said database.

44. (new) The method of claim 43, wherein said relaying said tracing token by said first interceptor comprises writing said tracing token to a shared memory location accessible by said second interceptor and said receiving said tracing token by said second interceptor comprises reading said tracing token at said shared memory location.

45. (new) The method of claim 43, wherein said first interceptor runs in a first process and said second interceptor runs in a second process, and said relaying said tracing token by said first interceptor and said receiving said tracing token by said second interceptor are accomplished using inter-process communication.

46. (new) A method of tracing application program execution paths comprising:  
providing a first interceptor and a second interceptor on an execution path, wherein said first interceptor is a first interception point on said execution path;  
generating a tracing token at said first interceptor, wherein said tracing token uniquely identifies said execution path;  
relaying said tracing token from said first interceptor to said second interceptor;  
collecting a first trace data at said first interceptor, wherein said first trace data includes said tracing token;  
collecting a second trace data at said second interceptor, wherein said second trace data includes said tracing token;  
correlating said first trace data with said second trace data using said tracing token; and  
reconstructing said execution path using said first trace data and said second trace data.

47. (new) The method of claim 46, wherein said relaying said tracing token from said first interceptor to said second interceptor is performed using shared memory.

48. (new) The method of claim 46, wherein said relaying said tracing token from said first interceptor to said second interceptor is performed using inter-process communication.

## Amendments to Claim:

This following listing of claims will replace all prior versions, and listings, of claims in the application.

## Listings of the Claims:

1. (currently amended) A computer implemented method for tracing an execution path through application components distributed over at least one computer in a data processing system, comprising the steps of:

providing a plurality of interceptors installed in a plurality of application components, wherein each application component has at least one interceptor;

receiving a plurality of requests to said data processing system, wherein processing each request by said data processing system produces an execution path traversing at least one application component;

generating a tracing token, activating data collection, and relaying said tracing token in one of said application components distributed over at least one computer; and

for a first interceptor, intercepting a first operation at a first application component;

for a first interceptor, determining if tracing an execution path associates with said first operation is desired;

for a first interceptor, if tracing said execution path associates with said first operation is desired, performing a first trace function, wherein said first trace function comprises:

generating a tracing token;

activating data collection at said first interceptor; and

relaying said tracing token;

detecting said tracing token, activating data collection, and relaying said tracing token in at least one of said application components;

for a second interceptor, intercepting a second operation at a second application component;

for a second interceptor, detecting said tracing token; and  
for a second interceptor, if said tracing token exists, performing a second  
trace function, wherein said second trace function comprises:  
activating data collection at said second interceptor; and  
relaying said tracking token.

2. (original) The method of claim 1, wherein said application components include at least one of application programs, plug-in programs, program extensions, static library programs, dynamic library programs, program scripts, and individual modules, or subroutines, or functions, or procedures, or methods thereof.

3. (original) The method of claim 1, wherein said application components reside in the same computer, or are distributed over multiple computers interconnected by at least one of a local area network, a wide-area network, a wireless network, or the Internet.

4. (currently amended) The method of claim 1, wherein ~~said generating a tracing token is controlled by~~ said determining if tracing an execution path associates with said first operation is desired utilizes a start rule and a stop rule.

5. (currently amended) The method of claim 1, wherein said tracing token ~~[[is]] comprises~~ a unique value in said data processing system, said unique value uniquely identifies a particular request.

6. (currently amended) The method of claim ~~[[1]]~~ 5, wherein said unique value ~~includes~~ is a universally unique identifier (UUID).

7. (original) The method of claim 1, wherein said relaying of said tracing token is accomplished through at least one of reading and writing of a shared memory, attachment to an inter-process message, and insertion into a data stream.

8. (currently amended) The method of claim ~~[[7]]~~ 1, wherein ~~said attachment to an inter-process message includes at least one of a non-intrusive modification of the message header, a non-intrusive modification of the message body, or~~

appending to the message said relaying said tracing token step at said first interceptor comprises inserting said tracing token in an inter-process message, said detecting said tracing token step at said second interceptor comprises reading said inter-process message to retrieve said tracing token, and said tracing token is detected when said tracing token is found in said inter-process message.

9. (currently amended) The method of claim [[8]] 1, wherein ~~said non-intrusive modification of the message header includes at least one of adding a new field in a SOAP transport header over HTTP, or adding a new property in a SOAP transport header over JMS~~ said relaying said tracing token step at said first interceptor comprises writing said tracing token to a data stream, said detecting said tracing token step at said second interceptor comprises reading said data stream to retrieve said tracing token, and said tracing token is detected when said tracing token is found in said data stream.

10. (currently amended) The method of claim [[8]] 1, wherein ~~said non-intrusive modification of the message body includes the insertion of a comment in an SQL statement~~ said relaying said tracing token step at said first interceptor comprises storing said tracing token in a shared memory location accessible by said second interceptor, said detecting said tracing token step at said second interceptor comprises reading said share memory location to locate said tracing token, and said tracing token is detected when said tracing token is found in said shared memory location.

11. (original) The method of claim 1, wherein said tracing an execution path through application components takes place in at least one of system production time, system development time, system integration time, or system testing time of said data processing system.

12. (original) The method of claim 1, wherein said generating a tracing token is automatically triggered by detection of a system event.

13. (original) The method of claim 12, wherein said detection of a system event is dynamically activated and deactivated.

Application Number 10/657,770

Page 4/19

~~... detecting said tracing token, wherein said relaying said tracing token is generated by~~  
said first monitoring code, and said second monitoring code is different from said first monitoring code;

detecting one of said system events, activating data collection, and generating and relaying one of said tracing tokens in one of said first plurality of said application components, wherein said one of said tracing tokens comprises a unique identifier uniquely identifies said execution path;

detecting and relaying said generated one of said tracing tokens to activate data collection in at least one of said second plurality of said application components along said execution path through said application components in said data processing system; and

recording said collected data [[in]] from all said application components along said execution path through said application components in said data processing system.

Application Number 10/657,770

Page 5/19

Application Number 10/657,770

Page 9/19

~~appending to the message~~ said relaying said tracing token step at said first interceptor comprises inserting said tracing token in an inter-process message, said detecting said tracing token step at said second interceptor comprises reading said inter-process message to retrieve said tracing token, and said tracing token is detected when said tracing token is found in said inter-process message.

9. (currently amended) The method of claim [[8]] 1, wherein ~~said non-intrusive modification of the message header includes at least one of adding a new field in a SOAP transport header over HTTP, or adding a new property in a SOAP transport header over JMS~~ said relaying said tracing token step at said first interceptor comprises writing said tracing token to a data stream, said detecting said tracing token step at said second interceptor comprises reading said data stream to retrieve said tracing token, and said tracing token is detected when said tracing token is found in said data stream.

10. (currently amended) The method of claim [[8]] 1, wherein ~~said non-intrusive modification of the message body includes the insertion of a comment in an SQL statement~~ said relaying said tracing token step at said first interceptor comprises storing said tracing token in a shared memory location accessible by said second interceptor, said detecting said tracing token step at said second interceptor comprises reading said share memory location to locate said tracing token, and said tracing token is detected when said tracing token is found in said shared memory location.

11. (original) The method of claim 1, wherein said tracing an execution path through application components takes place in at least one of system production time, system development time, system integration time, or system testing time of said data processing system.

12. (original) The method of claim 1, wherein said generating a tracing token is automatically triggered by detection of a system event.

13. (original) The method of claim 12, wherein said detection of a system event is dynamically activated and deactivated.

14. (original) The method of claim 13, wherein said dynamic activation and deactivation of said detection of a system event are controlled internally in said one of said application components, or externally to said one of said application components.

15. (original) The method of claim 12, wherein said system event includes at least one of an event relating to system and application performance, system and application availability, system and application troubleshooting, or system and application security.

16. (currently amended) A computer implemented method for tracing an execution path through application components distributed over at least one computer in a data processing system, comprising the steps of:

installing first monitoring code in a first plurality of said application components for detecting system events;

installing second monitoring code in a second plurality of said application components for detecting tracing tokens, wherein said tracing tokens are generated by said first monitoring code, and said second monitoring code is different from said first monitoring code;

detecting one of said system events, activating data collection, and generating and relaying one of said tracing tokens in one of said first plurality of said application components, wherein said one of said tracing tokens comprises a unique identifier uniquely identifies said execution path;

detecting and relaying said generated one of said tracing tokens to activate data collection in at least one of said second plurality of said application components along said execution path through said application components in said data processing system; and

recording said collected data [[in]] from all said application components along said execution path through said application components in said data processing system.



17. (original) The method of claim 16, wherein said first plurality of said application components includes entry points for client enquiries to an n-tier client/server distributed computing system.

18. (original) The method of claim 16, further comprising the step of providing storage means for said collected data for all said application components along said execution path through said application components in said data processing system.

19. (currently amended) The method of claim 18, wherein said storage means includes at least one of files, registries, relational databases, [[and]] or object databases.

20. (currently amended) The method of claim 16, wherein a record of said collected data includes at least one of:

- the value of one of said tracing tokens;
- the value of a computer's timestamp;
- the identifier of a computer;
- the identifier of one of said application components;
- the description of one of said system events; or
- application data from operation.

21. (currently amended) The method of claim 16, wherein said monitoring code is installed in at least one of [[the]] source code, [[the]] object code, [[the]] executable code, [[and]] or application extension mechanism of said application components.

22. (currently amended) A computer program product comprising~~[[ ]]~~ a computer usable medium having computer readable program code means embodied therein for tracing an execution path through application components distributed over at least one computer in a data processing system, the computer readable program code means in said computer program product comprising:

computer readable program code means for installing first monitoring code in a first plurality of said application components for detecting system events;